An ISS White Paper

Internet Scanner® 7.0
Technical Overview

6303 Barfield Road • Atlanta, GA 30328 Tel: 404.236.2600 • Fax: 404.236.2626

Overview

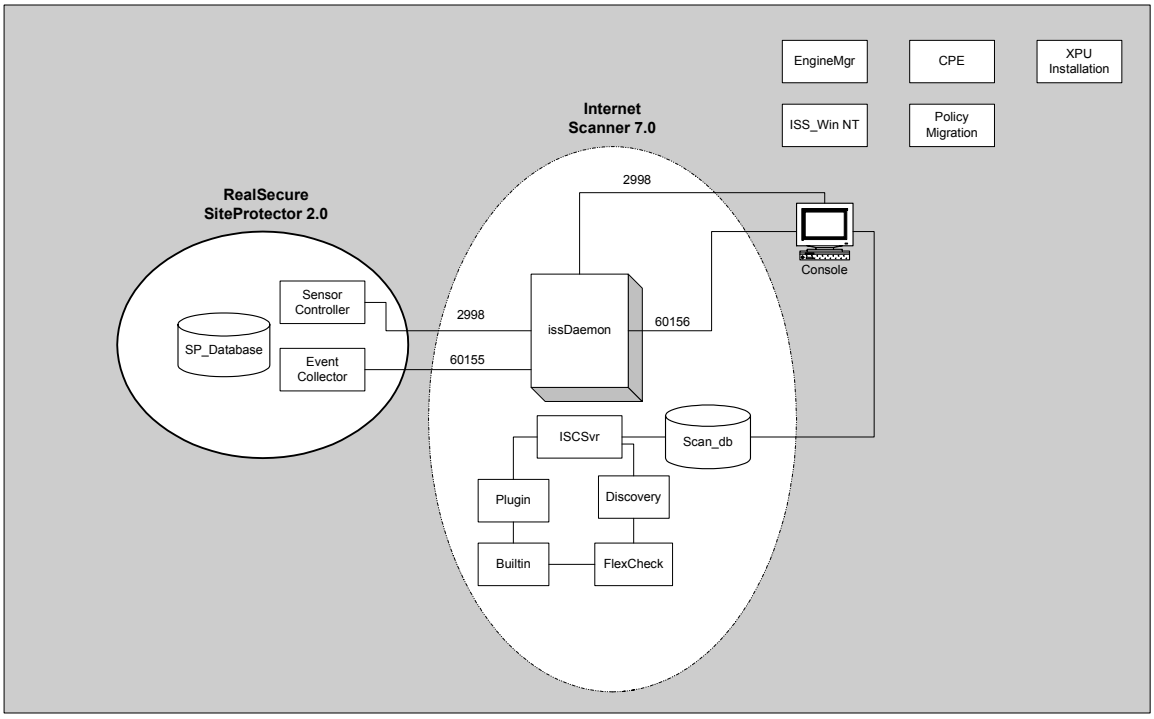
This document outlines the design and operation of Internet Scanner 7.0. Significant changes were made to the 7.0 version in an effort to increase performance, accuracy, stability and usability. The benefits resulting from these changes are explored here in this document as well.

The major new features covered here are:

- Architecture
  - Scanner Sensor
  - Communications
- TCP/IP Stack Fingerprinting
- Database
- SiteProtector Integration

Architecture

One of the most significant differences in Internet Scanner 7.0 is the architecture. The new communications architecture, similar to that of RealSecure® Network Sensor and RealSecure® Server Sensor, uses a client-server paradigm. The design of the scanner itself has changed as well, using a modular design that is much more extensible than the monolithic architecture used in previous versions. The following is a high-level diagram of the 7.0 scanner.



The components in the diagram are defined in the following sections.

Scanner Sensor

The Scanner Sensor includes the following services:

- issDaemon
- Internet Scanner Controller

issDaemon

The ISS Daemon provides a generic communication interface for the native console, a SiteProtector™ console or the command-line interface. Command and control functions sent through this component include:

- Restart Sensor
- Start scan
- Stop scan
- Install X-Press Update

The service added to the Windows Service Control Manager is called ***issDaemon***. Processes started by this service are:

- issDaemon.exe
- issCSF.exe

When started, issDaemon.exe listens for command and control connections on TCP port 2998. The issCSF.exe process listens on TCP ports 60155 for the creation of the event channel to RSSP and 60166 for the creation of the event channel to the Scanner Console. The purpose of these open ports is explained in a subsequent section. The port numbers for these services can be modified.

It is also important to note that the issDaemon service starts and stops the Internet Scanner Controller service, which is described in the next section. These services can, however, be stopped and started independently of each other.

Internet Scanner Controller

The other service added is the ***Internet Scanner Controller***. The Internet Scanner Controller (ISC), ISCSrv.exe, is responsible for directing the sub-processes that perform various scanning duties. In addition to task scheduling, the ISC manages the requests and responses coming to and from each sub-process. These sub-processes, also known as MicroEngines, are:

- Built-in Engine
- Plug-in Engine
- Discovery Engine
- FlexCheck Engine

Built-in Engine

The Built-in Engine loads and manages the execution of built-in vulnerability checks. These are checks that were created before the implementation of Internet Scanner's Plug-In / Built-in architecture. The primary difference between the two types of checks is that the built-in checks have resources that are embedded in the exploits, resulting in dependency relationships between some exploits.

Plug-in Engine

The Plug-in Engine loads and manages the execution of plug-in vulnerability checks. Plug-ins are autonomous modules that perform vulnerability check against a target host. Unlike built-in checks, plug-ins do not have any dependencies on other checks.

Discovery Engine

The Discovery Module is responsible for gathering identification information from hosts. This module includes the following sub-components:

- Fingerprinter
- ICMP pinger
- TCP pinger
- TCP port scanner
- UDP port scanner
- DNS lookup utility
- NetBIOS utilities
- Operating System Identification (OSID)

The fingerprinting component is new to Internet Scanner 7.0. It is responsible for sending the specially-crafted TCP packets used for fingerprint identification. There is a section on TCP/IP stack fingerprinting later in this document.

In addition to stack fingerprinting, Internet Scanner 7.0 relies on other methods, such as banner grabbing and NetBIOS queries, to determine the target host's operating systems with a high degree of confidence. The Discovery Engine uses all or a subset - depending on the policy - of the sub-components listed above to gather pieces of information that help to identify the target host. The Engine process then listens for responses and adds data that it has received into a host knowledge base (HKB). The HKB caches information about a host to improve performance during the scan. Host information is newly obtained for each scan; cached information from previous scans is not reused.

Internet Scanner 7.0 also includes the ability to perform TCP SYN, or "half-open", scans. By default, a full TCP connection is attempted on each specified port. TCP SYN scans are faster, but may not be as reliable.

Flex Check Engine

The Flex Check engine loads and executes FlexChecks™. FlexChecks are external programs that attempt to identify specific vulnerabilities on a host.

Additional sub-components that are part of Micro Engines include:

Exploit Manager

Checks are represented in the Exploit Manager as exploit objects. The Exploit Manager maintains a collection of exploit objects and exposes an interface to retrieve references to these objects.

Resource Manager

The Resource Manager maintains a list of network scanning resources, the namespace scope the resource lives in and its activation lifetime. A resource can either be a TCP connection, an FTP client, a password list an RPC connection or other resource utilized by an exploit. Internet Scanner uses the Resource Manager to provide uniform access to all types of objects regardless of their implementation.

Communications

The following table summaries Internet Scanner communications. These port values can be changed by modifying values in the files specified below.

Internet Scanner Communications			
Client	**TCP Port**	**Function**	**Port Specified In**
SiteProtector, Native console or CLI	2998	Command and control	\Program Files\ISS\issDaemon\issDaemon.policy [\config\]; daemonport =L 2998;
SiteProtector Event Collector	60155	Event and status data	\Program Files\ISS\issSensors\<scanner name>\common.policy [\Response\DISPLAY\Default\]; EngineListenPort =L 60155;
Native Internet Scanner console	60156	Event channel	\Program Files\ISS\issSensors\<scanner name>\common.policy [\Response\DISPLAYNP\Default\]; EngineListenPort =L 60156;

Encryption

Encryption functions use the Microsoft Cryptographic API. This allows the best encryption available on your system (40-bit to 128-bit symmetric encryption, 1024-bit or 1536-bit public key encryption) to be used. A different cryptographic provider can be used; however, users must install the provider prior to the installation of Internet Scanner. Communications are authenticated with a public-private key exchange algorithm and verified with cryptographic checksums appended and checked for each message.

TCP/IP Stack Fingerprinting

TCP/IP Stack Fingerprinting has been implemented in 7.0 to improve the accuracy of its Operating System Identification. This is an active fingerprinting technique whereby specially crafted TCP packets are sent to the target host. The responses sent by the target are compared to responses listed in a database.

The following is the basic process that is used by Internet Scanner to identify a host's operating system: First, the scanner attempts to identify at least one open and one closed port. Next, the fingerprint component sends a sequence of nine "tests." These tests are specially crafted TCP packets that will elicit responses from the target. These responses are then compared to entries in the Nmap fingerprint database to determine if there is a match. More information can be found at: http://www.insecure.org/nmap/nmap-fingerprinting-artical.html

NMAP Fingerprint Database

ISS has license NMAP's fingerprint database for use in Internet Scanner 7.0, as it is the most comprehensive fingerprint database available. At this time, it is not possible to add to additional fingerprints to the database, due to the fact that the database file's integrity is protected by a digital signature.

Database

The Microsoft Access database backend used in previous versions of Internet Scanner has been replaced with the Microsoft Desktop Engine (MSDE). The main benefits of this change include:

- Increased stability
- More robust back-end
- Improved scalability

Like 6.2.1, Internet Scanner 7.0 communicates with the database through an ODBC connection. Currently, a remote database is not supported, therefore, MSDE must be used if Internet Scanner is installed on a supported platform(i.e. Windows 2000 Professional or XP).

More information, such as the database schema, can be found in the Internet Scanner 7.0 User Guide:

http://documents.iss.net/literature/InternetScanner/IS_UG_7.0.pdf

SiteProtector 2.0 Integration

The client-server architecture used in Internet Scanner 7.0 allows for native SiteProtector 2.0 support. This means that the scanner no longer requires a Databridge to send vulnerability data and host information back to SiteProtector. In addition, Internet Scanner sensors managed by SiteProtector can be installed in a "headless" manner (i.e. without a console). In this configuration, the scanner is managed by the SiteProtector console and the data it collects is sent to a SiteProtector Event Collector.

When a scan is initiated, a request is sent from the console to the Scanner Sensor Controller, which delegates tasks to the Micro Engines. Scan data is temporarily stored in a local event queue until the Event Collect confirms that the data has been committed to the database.

Audit Scanning

Internet Scanner 7.0 can be used in an audit capacity even if it is primarily managed by SiteProtector. This is due the separate event channels available for SiteProtector and native console. When a scan is initiated by the native console, events (vulnerability and host information) are stored in the local event queue mentioned in the previous section. When an authorized Event Collector connects to the scanner sensor, the data is sent in a normal manner. The local event queue has a default size of 15 megabytes. This value can be increased or decreased by modifying the following line in the *\Program Files\ISS\issSensors\<sensor name>\common.policy* file.

SensorEventQueueSite =L 1500000;

By default, once the queue reaches its maximum size, events will no longer be logged. In this situation, the user must empty the queue file by deleting it, clearing it with the ADF Queue Maintenance utility or allowing an Event Collector to connect to the sensor.

About Internet Security Systems (ISS)
Founded in 1994, Internet Security Systems (ISS) (Nasdaq: ISSX) is a pioneer and world leader in software and services that protect corporate and personal information from an ever-changing spectrum of online threats and misuse. Internet Security Systems is headquartered in Atlanta, GA, with additional operations throughout the Americas, Asia, Australia, Europe and the Middle East. For more information, visit the Internet Security Systems Web site at www.iss.net or call 888-901-7477.